Lab Exercise 1 Revision 2

Due: April 19, 2021   **Points:** 100

<div style="color:red; text-align:center; border:1px solid red; display:inline-block">
Revision 2 has changed the user "sushi" to "sshd" in the second example.
</div>

<div style="color:red; text-align:center; border:1px solid red; display:inline-block">
Revision 1 has the submission instructions added.
</div>

This laboratory exercise is designed to teach you about Linux file permissions.

You are to write a program called "access" that will say what rights a user or group has over a file or directory.

Command

The command for your program is the following:

```
access[ -g ] name  file1 ...
```

In this command, *name* is either a name or a non-negative integer. If the `-g` flag is given, *name* is interpreted as a group name or GID, and if that flag is not given, it is interpreted as a user name or UID. It is an error to give a non-existent user or group name, but it is *not* an error to give a UID or GID without an associated name.

Example Output

In these examples, "files" is a file and "dirt" is a directory. Assume "filex" has owner bishop, group owner users, and protection mode 0654, and the directory "dirx" has the same owner and group and the protection mode 0751. Then the command

```
access bishop filex
```

prints

```
The user bishop (UID 7000403) can read and write the file filex
```

The command

```
access sshd filex
```

prints

```
The user sshd (UID 74) can read the file filex
```

and the command

```
access 10 filex
```

prints

```
The user with UID 10 can read the file filex
```

but the command

```
access -g 10 filex
```

prints

```
Members of the group users (GID 10) can read and execute the file filex
```

The difference is that, without the `-g` option, the 10 is interpreted as a UID, and there is no associated name. But with that option, the 10 is interpreted as a GID, and that has the associated name "users".

If any of the named files are directories, you are to print "list the contents of", "modify", or "search" rather than "read", "write", and "execute", respectively. So, the command

```
access bishop dirx
```

prints

```
The user bishop (UID 7000403) can list the contents of, modify, and search the directory
                                         dirx
```

(all on the same line), the command

```
                         access -g users dirx
```

prints

```
 Members of the group users (GID 10) can list the contents of and search the directory dirx
```

and the command

```
                         access -g cosmos dirx
```

prints

```
          Members of the group cosmos (GID 900) can search the directory dirx
```

Checking Your Output

A reference version of this program is available in the directory ∼bishop/ecs153 on the CSIF; it is the executable file "access". Your output, and especially your error output, is to match that of this program.

Submitting Your Program

Please turn in the source code, the *Makefile* (you must include one) and any related information (such as manual pages and README files). Put these in a directory called *lab1*, create a *tar*(1) file called *lab1.tar*, and submit that on Canvas.

We will grade your program on the CSIF, so be sure it runs there. To grade it, we will unpack the tar file, run *make*, and then run some tests. Your *Makefile* must create an executable called "access" as that is the command we will use to run the tests.

If you use an interpreted language such as python, be sure you make the script executable in such a way that it can be executed by invoking it as "access".